Preface

Utility computing is a recent development in IT (information technology) outsourcing, whereby service capacity is provided as needed and the customer pays only for actual use. To provide what amounts to computing on demand, providers face business and technological challenges, including pricing of new services, adopting new business models, automated management of resource provisioning, and management of service level agreements (SLAs).

This issue of the *IBM Systems Journal* contains twelve papers on a wide range of topics related to utility computing. We have organized the papers into three groups: the first three papers deal with the business of utility computing, the next three describe the design or implementation of specific utility computing services, and the remaining six papers address issues of methodology and infrastructure.

In their paper, "Preparing for utility computing: The role of IT architecture and relationship management," Ross and Westerman examine the likely impact of utility computing on IT outsourcing. Drawing on a study of outsourcing involving eleven firms, they determine that the benefits of using new utility computing services will depend for the most part on how successfully firms manage their vendor relationships and on the capabilities and maturity of a firm's IT architecture. They make a number of recommendations to firms for managing their outsourcing relationships and for evolving their IT architectures in order to capitalize on the strategic advantages that utility computing may deliver.

In "Price-at-risk: A methodology for pricing utility computing services," Paleologo addresses the problem of pricing a service with on demand attributes and proposes a novel methodology, Price-at-Risk, that explicitly takes into account uncertainty in the pricing decision. By explicitly modeling contingent factors, such as the rate of service adoption or demand elasticity, the methodology can account for risk before the pricing decision is made.

Utility computing services have much in common with public service utilities. In "The utility business model and the future of computing services," Rappa compares the utility business model with the model for public utilities and discusses its potential role in future computing services.

Over the past few years, from its appearance in the marketplace until now, the business of "content distribution" has experienced rapid growth. A service to Web site owners that originally focused on improving customers' access times to Web sites, content distribution has evolved into an array of services that includes outsourcing of resources, improving Web site security, and support for multimedia streaming. In "A Web content serving utility," Gayek et al. describe the development and the resulting performance of a highly scalable content-serving utility computing system.

WebFountain, described by Gruhl et al. in "How to build a WebFountain: An architecture for very large-scale text analytics," is a platform for very large scale text analytics applications. It operates as a back-end supercomputing cluster and is offered to IBM partners as a utility computing service that they can use to build services for their customers. WebFountain is now operational and powers services, such as corporate reputation management systems and applications for preventing money laundering, which are offered by IBM partners to their clients.

In the paper "An architecture for the coordination of system management services," Naik, Mohindra, and Bantz describe an architecture for a "meta-management" service that is based on the utility computing model and can perform systems management tasks involving a set of distributed non-homogeneous systems management components. This new utility computing model is intended to broaden the scope of current systems management services to multicustomer utility computing environments while reducing the cost of providing these services.

In their paper "Using a utility computing framework to develop utility systems," Eilam et al. describe a utility computing framework that consists of a component model, a methodology, and a set of tools and common services, and that is used to build utility computing systems. The paper demonstrates the benefits of the framework by describing two implementations: a life-science utility computing service designed and implemented using the framework, and a partially implemented on-line gaming service designed in compliance with the framework.

"Policy-based computing" refers to a way of managing the operations of computer systems by specifying objectives to be met rather than the procedure to be executed (i.e., specifying "what" is to be achieved rather than "how" to achieve it). The use of policy-based computing leads to the simplification and automation of the tasks involved in operating computer systems. Appleby et al. describe in "Policy-based automated provisioning" the application of policy-based computing to the automatic provisioning of resources in the utility computing framework (the topic of the paper by Eilam et al.).

The Emerging Technologies Toolkit (ETTK) provides a run-time environment, as well as demos, examples, and additional tools that can be used to experiment with new technologies. The methodology behind the utility computing services track of ETTK is described in the paper "Web services on demand: WSLA-driven automated management" by Dan et al. The paper describes a methodology and a specification language, the Web Service Level Agreement language, for the automated management of SLAs.

In "Utility computing SLA management based upon business objectives," Buco et al. describe an SLA management system that can be integrated into a utility computing operating environment. Preliminary results from simulations and an early pilot implementation show this business-oriented design is likely to reduce the financial risks associated with service level violations.

The Universal Management Infrastructure (UMI) is an architecture for common functions to be used in the implementation of various utility computing services. In "Utility Metering Service in UMI," Albaugh and Madduri describe the metering function within UMI, which includes collecting metered data, storing data, calculating service metrics, and feeding these metrics to various consumer modules (e.g., accounting and billing).

Some believe that many software applications will soon be offered as utility computing services. In their paper "Design of an enablement process for on demand applications" Chang et al. describe a process, the Application Enablement Program, for transforming applications into utility computing services.

We thank our guest editors Dilip Kandlur and John Killela for their indispensable contribution to the planning and the production of this issue.

The next issue of the *Journal* is devoted to the IBM WebSphere Application Server.

Alex Birman, Associate Editor
John J. Ritsko, Editor-in-Chief